Notes for AA214, Chapter 12
SPLIT AND FACTORED FORMS

T. H. Pulliam

Stanford University

Introduction

1. Factored forms of numerical operators are used extensively in constructing and applying numerical methods to problems in fluid mechanics.

2. Concepts such as: "hybrid", "time split", and "fractional step" methods.

3. Especially useful for the derivation of practical algorithms that use implicit methods.

Concepts

1. Concept 1: Matrices can be split in quite arbitrary ways.
2. Concept 2: Advancing to the next time level always requires some reference to a previous one.
3. Concept 3: Time marching methods are valid only to some order of accuracy in the step size, h.

Splitting

1. Starting with:

$$\frac{d\vec{u}}{dt} = A\vec{u} - \vec{f} \tag{1}$$

2. Consider arbitrary splitting of A, Concept 1:

$$\frac{d\vec{u}}{dt} = [A_1 + A_2]\vec{u} - \vec{f} \tag{2}$$

where $A = [A_1 + A_2]$, but A_1 and A_2 are not unique.

1^{st} order Explicit Splitting

1. Choose the simple, first-order,[a] explicit Euler method.
2. New data $\vec{u}_{n+1}$ in terms of old $\vec{u}_n$ Concept 2:

$$\vec{u}_{n+1} = [\, I \ + hA_1 + hA_2]\vec{u}_n - h\vec{f} + O(h^2) \tag{3}$$

3. Equivalently:

$$\vec{u}_{n+1} = [[\, I \ + hA_1][\, I \ + hA_2] - h^2 A_1 A_2]\vec{u}_n - h\vec{f} + O(h^2)$$

[a] Second-order time-marching methods are considered later.

4. Finally, from Concept 3: (allowing us to drop higher order terms):

$$\vec{u}_{n+1} = [\, I \ + hA_1][\, I \ + hA_2]\vec{u}_n - h\vec{f} + O(h^2) \tag{4}$$

5. Eqs. 3 and 4 have the same formal order of accuracy

6. Neither one is to be preferred over the other.

7. However, their numerical stability can be quite different

8. Also, techniques to carry out their numerical evaluation can have arithmetic operation counts that vary by orders of magnitude.

Factoring Physical Representations: Time Splitting

1. PDE representing: convection A_c and dissipation A_d .

$$\frac{d\vec{u}}{dt} = A_c\vec{u} + A_d\vec{u} + (\vec{bc}) \tag{5}$$

2. Euler Explicit

$$\vec{u}_{n+1} = [\, I \; + hA_d + hA_c]\vec{u}_n + h(\vec{bc}) + O(h^2) \tag{6}$$

3. Factoring the dissipation term and the convection term produces a two step process

4. Results in additional error and possible stability consequences.

5. Factored form

$$
\begin{aligned}
\vec{u}_{n+1} &= [\,I + hA_d]\Big([\,I + hA_c]\vec{u}_n + h(\vec{bc})\Big) \\
&= \underbrace{[\,I + hA_d + hA_c]\vec{u}_n + h(\vec{bc})}_{\text{Original Unfactored Terms}} \\
&+ \underbrace{h^2 A_d\Big(A_c\vec{u}_n + (\vec{bc})\Big)}_{\text{Higher-Order Terms}} + O(h^2)
\end{aligned} \tag{7}
$$

6. Eq. 7 and the original unfactored form Eq. 6 have identical orders of accuracy in the time approximation.

7. Apply a predictor-corrector sequence.

$$
\begin{aligned}
\tilde{u}_{n+1} &= [\,I + hA_c]\vec{u}_n + h(\vec{bc}) \\
\vec{u}_{n+1} &= [\,I + hA_d]\tilde{u}_{n+1}
\end{aligned} \tag{8}
$$

Implicit-Explicit Factoring

1. Split combinations of implicit and explicit techniques.
2. Apply a partially implicit-explicit method to Eq. 5
$$\vec{u}_{n+1} = [\,I \; + hA_c]\vec{u}_n + hA_d\vec{u}_{n+1} + h(\vec{bc}) + O(h^2) \tag{9}$$
3. Rewrite as:
$$\begin{aligned}\vec{u}_{n+1} \quad &= \quad [\,I \; - hA_d]^{-1}\Big([\,I \; + hA_c]\vec{u}_n + h(\vec{bc})\Big) \\ &= \quad \underbrace{[\,I \; + hA_d + hA_c]\vec{u}_n + h(\vec{bc})}_{\text{Original Unfactored Terms}} + O(h^2)\end{aligned} \tag{10}$$
4. Using
$$[\,I \; - hA_d]^{-1} = I + hA_d + h^2A_d^2 + \cdots$$
if $h \cdot ||A_d|| < 1$, where $||A_d||$ is some norm of $[A_d]$.

5. A predictor-corrector interpretation leads to the sequence

$$\begin{aligned} \tilde{u}_{n+1} &= [\,I + hA_c]\vec{u}_n + h(\vec{bc}) \\ [\,I - hA_d]\vec{u}_{n+1} &= \tilde{u}_{n+1} \end{aligned} \tag{11}$$

6. The diffusion operator is now implicit, requiring a tridiagonal solver if the diffusion term is central differenced.

7. Numerical stiffness is generally much more severe for the diffusion process, this factored form would appear to be superior to that provided by Eq. 8. **But, Stability?**

Factoring Space Matrix Operators in 2–D

1. Physical systems are inherently multidimensional
2. Three-Dimensional (3D) Wave equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + c\frac{\partial u}{\partial z} = 0$$

3. Two-Dimensional (2D) Diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

4. Navier-Stokes equations

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

Mesh Indexing Convention

1. Linear 2-D scalar PDE that models diffusion:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{12}$$

2. Reduce PDE to a coupled set of ODE's by differencing the space derivatives on a mesh (graph, net).

3. Inspecting the resulting matrix operator.

Mesh Indexing

1. Assume a 3×4 point mesh[a]

$$
\begin{array}{cccccccc}
 & & \odot & \odot & \odot & \odot & \\
M_y & \odot & 13 & 23 & 33 & 43 & \odot \\
k & \odot & 12 & 22 & 32 & 42 & \odot \\
1 & \odot & 11 & 21 & 31 & 41 & \odot \\
 & & \odot & \odot & \odot & \odot & \\
 & & 1 & j & \cdots & M_x &
\end{array}
\tag{13}
$$

Mesh indexing in 2-D.

[a]This could also be called a 5×6 point mesh if the boundary points (labeled $\odot$ in the sketch) were included,here we just consider interior points.

2. $M_x = 4$, the number of (interior) x points

3. $M_y = 3$, the number of (interior) y points

4. The numbers $11\,,\ 12\,,\ \cdots\,,\ 43$ represent the location in the mesh of the dependent variable bearing that index.

5. Thus u_{32} represents the value of u at $j = 3$ and $k = 2$.

6. *NOTE:* This notation assume we order the points with the j or x index first and the k or y index second.

Data-Bases and Space Vectors

1. *DATA-BASE*: dimensioned array in a computer code that allots the storage locations of the dependent variable(s)
2. Many ways to lay out a data-base.
3. Consider only two:
   (a) (x)-vectors: consecutively along rows that are themselves consecutive from $k = 1$ to M_y
   (b) (y)-vectors: Consecutively along columns that are consecutive from $j = 1$ to M_x.
4. Refer to each row or column group as a *space vector* (they represent data along lines that are continuous in space)
5. Label their sum with the symbol U.

6. To be specific about the structure

   (a) Label a data-base composed of x-vectors with $U^{(x)}$

   (b) Example:
   $U^{(x)} = (u_{11}, u_{21}, u_{31}, u_{41}, u_{12}, u_{22}, u_{32}, u_{42}, u_{13}, u_{23}, u_{33}, u_{43})^T$

   (c) Label a data-base composed of y-vectors with $U^{(y)}$.

   (d) Example:
   $U^{(y)} = (u_{11}, u_{12}, u_{13}, u_{21}, u_{22}, u_{23}, u_{31}, u_{32}, u_{33}, u_{41}, u_{42}, u_{43})^T$

Data-Base Permutations

1. Two vectors (arrays) are related by a permutation matrix P:

$$U^{(x)} = P_{xy} U^{(y)} \quad \text{and} \quad U^{(y)} = P_{yx} U^{(x)} \quad \text{with} \quad P_{yx} = P_{xy}^{T} = P_{xy}^{-1} \tag{14}$$

2. Consider the structure of a matrix finite-difference operator representing 3-point central-differencing schemes for both space derivatives in two dimensions.

3. When the matrix is multiplying a space vector U, the usual (but ambiguous) representation is given by A_{x+y}.

$$\frac{dU}{dt} = A_{x+y} U + (\vec{bc}) \tag{15}$$

4. If it is important to be specific about the data-base structure, we use the notation $A_{x+y}^{(x)}$ or $A_{x+y}^{(y)}$, depending on the data-base chosen for the U it multiplies.

Example for $U^{(x)}$ Ordering

1. Data-base composed of M_y x–vectors stored in $U^{(x)}$.

2. Entries for $x \to x$, for $y \to o$, for both $\to \bullet$.

$$
A^{(x)}_{x+y} \cdot U^{(x)} = \left[\begin{array}{cccc|cccc|cccc}
\bullet & x & & & o & & & & & & & \\
x & \bullet & x & & & o & & & & & & \\
& x & \bullet & x & & & o & & & & & \\
& & x & \bullet & & & & o & & & & \\
& & & & & & & & & & & \\
o & & & & \bullet & x & & & o & & & \\
& o & & & x & \bullet & x & & & o & & \\
& & o & & & x & \bullet & x & & & o & \\
& & & o & & & x & \bullet & & & & o \\
& & & & & & & & & & & \\
& & & & o & & & & \bullet & x & & \\
& & & & & o & & & x & \bullet & x & \\
& & & & & & o & & & x & \bullet & x \\
& & & & & & & o & & & x & \bullet
\end{array}\right] \cdot \begin{array}{c}
11 \\ 21 \\ 31 \\ 41 \\ -\,- \\ 12 \\ 22 \\ 32 \\ 42 \\ -\,- \\ 13 \\ 23 \\ 33 \\ 43
\end{array} \tag{16}
$$

Example for $U^{(y)}$ Ordering

1. Data-base composed of M_x y–vectors stored in $U^{(y)}$.

2. Entries for $x \to x$, for $y \to o$, for both $\to \bullet$.

$$A^{(y)}_{x+y} \cdot U^{(y)} = \left[\begin{array}{ccc|ccc|ccc|ccc}
\bullet & o & & x & & & & & & & & \\
o & \bullet & o & & x & & & & & & & \\
& o & \bullet & & & x & & & & & & \\
& & & & & & & & & & & \\
x & & & \bullet & o & & x & & & & & \\
& x & & o & \bullet & o & & x & & & & \\
& & x & & o & \bullet & & & x & & & \\
& & & & & & & & & & & \\
& & & x & & & \bullet & o & & x & & \\
& & & & x & & o & \bullet & o & & x & \\
& & & & & x & & o & \bullet & & & x \\
& & & & & & & & & & & \\
& & & & & & x & & & \bullet & o & \\
& & & & & & & x & & o & \bullet & o \\
& & & & & & & & x & & o & \bullet
\end{array}\right] \cdot \begin{array}{c}
11 \\ 12 \\ 13 \\ -- \\ 21 \\ 22 \\ 23 \\ -- \\ 31 \\ 32 \\ 33 \\ -- \\ 41 \\ 42 \\ 43
\end{array} \tag{17}$$

Permutation Discussion

1. Notice that the matrices are not the same although they represent the same derivative operation.

2. Their structures are similar, however, and they are related by the same permutation matrix that relates $U^{(x)}$ to $U^{(y)}$.

$$A^{(x)}_{x+y} = P_{xy} \cdot A^{(y)}_{x+y} \cdot P_{yx} \tag{18}$$

Space Splitting and Factoring: (x) Ordering

1. The matrix $A^{(x)}_{x+y}$ can be split into two matrices such that

$$A^{(x)}_{x+y} = A^{(x)}_{x} + A^{(x)}_{y} \tag{19}$$

$$A^{(x)}_{x} \cdot U^{(x)} = \left[\begin{array}{cccc|cccc|cccc}
x & x & & & & & & & & & & \\
x & x & x & & & & & & & & & \\
& x & x & x & & & & & & & & \\
& & x & x & & & & & & & & \\
& & & & x & x & & & & & & \\
& & & & x & x & x & & & & & \\
& & & & & x & x & x & & & & \\
& & & & & & x & x & & & & \\
& & & & & & & & x & x & & \\
& & & & & & & & x & x & x & \\
& & & & & & & & & x & x & x \\
& & & & & & & & & & x & x
\end{array}\right] \cdot U^{(x)} \tag{20}$$

$$A_y^{(x)} \cdot U^{(x)} = \left[\begin{array}{cccc|cccc|cccc}
o & & & & o & & & & & & & \\
& o & & & & o & & & & & & \\
& & o & & & & o & & & & & \\
& & & o & & & & o & & & & \\
o & & & & o & & & & o & & & \\
& o & & & & o & & & & o & & \\
& & o & & & & o & & & & o & \\
& & & o & & & & o & & & & o \\
& & & & o & & & & o & & & \\
& & & & & o & & & & o & & \\
& & & & & & o & & & & o & \\
& & & & & & & o & & & & o
\end{array}\right] \cdot U^{(x)} \qquad (21)$$

Space Splitting and Factoring: (y) Ordering

1. Similarly

$$A^{(y)}_{x+y} = A^{(y)}_{x} + A^{(y)}_{y} \tag{22}$$

$$A^{(y)}_{x} \cdot U^{(y)} = \left[\begin{array}{ccc|ccc|ccc|ccc}
x & & & x & & & & & & & & \\
& x & & & x & & & & & & & \\
& & x & & & x & & & & & & \\
x & & & x & & & x & & & & & \\
& x & & & x & & & x & & & & \\
& & x & & & x & & & x & & & \\
& & & x & & & x & & & x & & \\
& & & & x & & & x & & & x & \\
& & & & & x & & & x & & & x \\
& & & & & & x & & & x & & \\
& & & & & & & x & & & x & \\
& & & & & & & & x & & & x
\end{array}\right] \cdot U^{(y)} \tag{23}$$

$$A_y^{(y)} \cdot U^{(y)} = \left[\begin{array}{ccc|ccc|ccc|ccc}
o & o & & & & & & & & & & \\
o & o & o & & & & & & & & & \\
& o & o & & & & & & & & & \\
& & & o & o & & & & & & & \\
& & & o & o & o & & & & & & \\
& & & & o & o & & & & & & \\
& & & & & & o & o & & & & \\
& & & & & & o & o & o & & & \\
& & & & & & & o & o & & & \\
& & & & & & & & & o & o & \\
& & & & & & & & & o & o & o \\
& & & & & & & & & & o & o
\end{array}\right] \cdot U^{(y)} \qquad (24)$$

Matrix Relations: Permutations

1. The permutation relation also holds for the split matrices, so

$$A_y^{(x)} = P_{xy} A_y^{(y)} P_{yx}$$

and

$$A_x^{(x)} = P_{xy} A_x^{(y)} P_{yx}$$

2. The splittings can be combined with factoring

Example

1. First-order in time:implicit Euler method

$$U_{n+1}^{(x)} = U_n^{(x)} + h\left[A_x^{(x)} + A_y^{(x)}\right]U_{n+1}^{(x)} + h(\vec{bc})$$

$$\left[I - hA_x^{(x)} - hA_y^{(x)}\right]U_{n+1}^{(x)} = U_n^{(x)} + h(\vec{bc}) + O(h^2) \tag{25}$$

2. Retain first-order accuracy with the alternative

$$\left[I - hA_x^{(x)}\right]\left[I - hA_y^{(x)}\right]U_{n+1}^{(x)} = U_n^{(x)} + h(\vec{bc}) + O(h^2) \tag{26}$$

3. Predictor-corrector form and permute the data-base of the second row.

$$\begin{aligned} \left[I - hA_x^{(x)}\right]\tilde{U}^{(x)} &= U_n^{(x)} + h(\vec{bc}) \\ \left[I - hA_y^{(y)}\right]U_{n+1}^{(y)} &= \tilde{U}^{(y)} \end{aligned} \tag{27}$$

Second-Order Factored Implicit Methods

1. Second-order accuracy in time can be maintained

2. Trapezoidal method where the derivative operators have been split

$$\left[I - \frac{1}{2}hA_x - \frac{1}{2}hA_y\right]U_{n+1} = \tag{28}$$

$$\left[I + \frac{1}{2}hA_x + \frac{1}{2}hA_y\right]U_n + h(\vec{bc}) + O(h^3)$$

3. Factor both sides giving

$$\left[\left[I - \frac{1}{2}hA_x\right]\left[I - \frac{1}{2}hA_y\right] - \frac{1}{4}h^2A_xA_y\right]U_{n+1}$$

$$= \left[\left[I + \frac{1}{2}hA_x\right]\left[I + \frac{1}{2}hA_y\right] - \frac{1}{4}h^2A_xA_y\right]U_n +$$

$$h(\vec{bc}) + O(h^3) \tag{29}$$

4. Note that the combination $\frac{1}{4}h^2[A_x A_y](U_{n+1} - U_n)$ is proportional to h^3: since $(U_{n+1} - U_n)$ is proportional to h.

$$\begin{aligned}\left[I - \frac{1}{2}hA_x\right]\left[I - \frac{1}{2}hA_y\right]U_{n+1} &= \\ \left[I + \frac{1}{2}hA_x\right]\left[I + \frac{1}{2}hA_y\right]U_n &+ h(\vec{bc}) + O(h^3)\end{aligned}$$

5. Both the factored and unfactored form of the trapezoidal method are second-order accurate in the time march.

6. An alternative form of this kind of factorization is the classical ADI (alternating direction implicit) method

$$\begin{aligned}\left[I - \frac{1}{2}hA_x\right]\tilde{U} &= \left[I + \frac{1}{2}hA_y\right]U_n + \frac{1}{2}hF_n \\ \left[I - \frac{1}{2}hA_y\right]U_{n+1} &= \left[I + \frac{1}{2}hA_x\right]\tilde{U} + \frac{1}{2}hF_{n+1} + O(h^3)\end{aligned}$$

Importance of Factored Forms: 2D and 3D

1. Time-march equations are stiff, and implicit methods are required to permit reasonably large time steps

2. The use of factored forms becomes a very valuable tool for realistic problems.

3. Consider,

$$\left[I - \frac{1}{2}hA_{x+y}\right]U_{n+1} = \left[I + \frac{1}{2}hA_{x+y}\right]U_n + h(\vec{bc})$$

4. M_x number of points in x, M_y number of points in y

5. Accumulate the right-hand-side in the array (RHS).

6. Solving for U_{n+1} requires the solution of a sparse, but very large, set of coupled simultaneous equations, e.g.,

$$\left[I - \frac{1}{2} h A_{x+y}\right] \cdot U_{n+1} = \left[\begin{array}{cccc|cccc|cccc}
\bullet & x & & & o & & & & & & & \\
x & \bullet & x & & & o & & & & & & \\
& x & \bullet & x & & & o & & & & & \\
& & x & \bullet & & & & o & & & & \\
o & & & & \bullet & x & & & o & & & \\
& o & & & x & \bullet & x & & & o & & \\
& & o & & & x & \bullet & x & & & o & \\
& & & o & & & x & \bullet & & & & o \\
& & & & o & & & & \bullet & x & & \\
& & & & & o & & & x & \bullet & x & \\
& & & & & & o & & & x & \bullet & x \\
& & & & & & & o & & & x & \bullet
\end{array}\right] \cdot U_{n+1} = (RHS)$$

7. In real cases involving the 2-D Euler or Navier-Stokes equations, each symbol $(o, x, \bullet)$ represents a 4×4 block matrix with entries that depend on the pressure, density and velocity field, (5×5 in 3D).

8. Suppose we were to solve the equations directly.

9. The forward sweep of a simple Gaussian elimination fills all of the 4×4 blocks between the main and outermost diagonal,

10. This must be stored in computer memory to be used to find the final solution in the backward sweep.

11. If N_e represents the order of the small block matrix (4 in the 2-D Euler case), the approximate memory requirement is

$$(N_e \times M_y) \cdot (N_e \times M_y) \cdot M_x$$

floating point words.

12. Here it is assumed that $M_y < M_x$. If $M_y > M_x$, M_y and M_x would be interchanged.

13. A moderate mesh of 60×200 points would require over 11 million words to find the solution.

14. The next consideration is operation counts for the solution process.

15. A full matrix of rank (size) N requires $O(N^3)$ floating point operations (FLOP)to solve the linear system.

16. A matrix with band width b[a] requires $O(Nb^2)$ FLOP

17. The 2D example would then require
$O((N_e \cdot M_x \cdot M_y) \cdot Min(N_e \cdot M_x, N_e \cdot M_y)^2)$ FLOP

18. The 60×200 point example would require $\approx 10^9$ FLOP

19. In a practical application, each iteration n requires the large sparse linear solve with typically 10^3 to 10^4 times steps, leading to $\approx 10^{12} - 10^{13}$ FLOP per case.

20. With computing speeds of over one teraflop,[b] direct solvers may become useful for finding steady-state solutions of practical problems in two dimensions.

---

[a] Band width ,b, is defined as the maximum number of off diagonals with non-zero entries plus one for the center diagonal. For example: a tridiagonal matrix has $b = 2$, a pentra-diagonal matrix $b = 3$

[b] One trillion floating-point operations per second.

21. However, a three-dimensional solver would require a memory of approximately

$$N_e^2 \cdot M_y^2 \cdot M_z^2 \cdot M_x$$

words, and, for well resolved flow fields, this probably exceeds memory availability for some time to come.

22. Operation counts for a direct solver in 3D [c] are in the $> 10^{16}$ range

23. On the other hand, consider computing a solution using the *factored* implicit equation

$$\begin{aligned}\left[I - \frac{1}{2}hA_x\right]\left[I - \frac{1}{2}hA_y\right]U_{n+1} &= \\ \left[I + \frac{1}{2}hA_x\right]\left[I + \frac{1}{2}hA_y\right]U_n &+ h(\vec{bc})\end{aligned}$$

24. Again, form the (RHS)

[c] A 4 million point grid: $M_x = 100, M_y = 200, M_z = 200, N_e = 5$

25. Write the remaining terms in the two-step predictor-corrector form

$$\begin{aligned} \left[I - \frac{1}{2}hA_x^{(x)}\right]\tilde{U}^{(x)} &= (RHS)^{(x)} \\ \left[I - \frac{1}{2}hA_y^{(y)}\right]U_{n+1}^{(y)} &= \tilde{U}^{(y)} \end{aligned} \tag{30}$$

26. First step: solved using M_y uncoupled block tridiagonal solvers[d].

27. This is equivalent to solving M_y *one-dimensional, 1-D* problems, each with M_x blocks of order N_e.

$$\left[I - \frac{1}{2} h A_x^{(x)}\right] \tilde{U}^{(x)} = \left[\begin{array}{cccc|cccc|cccc} x & x & & & & & & & & & & \\ x & x & x & & & & & & & & & \\ & x & x & x & & & & & & & & \\ & & x & x & & & & & & & & \\ & & & & x & x & & & & & & \\ & & & & x & x & x & & & & & \\ & & & & & x & x & x & & & & \\ & & & & & & x & x & & & & \\ & & & & & & & & x & x & & \\ & & & & & & & & x & x & x & \\ & & & & & & & & & x & x & x \\ & & & & & & & & & & x & x \end{array}\right] \cdot \tilde{U}^{(x)} = (RHS)^{(x)}$$

28. Temporary solution $\tilde{U}^{(x)}$ would then be permuted to $\tilde{U}^{(y)}$

[d] A block tridiagonal solver is similar to a scalar solver except that small block matrix operations replace the scalar ones, and matrix multiplications do not commute.

29. Next step: solve M_x *1D* implicit problems each with dimension M_y.

$$\left[I - \frac{1}{2} h A_y^{(y)}\right] U_{n+1}^{(y)} = \left[\begin{array}{ccc|ccc|ccc|ccc} y & y & & & & & & & & & & \\ y & y & y & & & & & & & & & \\ & y & y & & & & & & & & & \\ & & & y & y & & & & & & & \\ & & & y & y & y & & & & & & \\ & & & & y & y & & & & & & \\ & & & & & & y & y & & & & \\ & & & & & & y & y & y & & & \\ & & & & & & & y & y & & & \\ & & & & & & & & & y & y & \\ & & & & & & & & & y & y & y \\ & & & & & & & & & & y & y \end{array}\right] \cdot U_{n+1}^{(y)} = \tilde{U}^{(y)}$$

30. The band width for both steps is now $b = 2N_e$

31. An Operation count for each step is: $M_y \times M_x \times 4 \times N_e^2$

32. In the 60×200 example the total is : $\approx 2 \times 10^6$ FLOP per time step

33. Significantly less than the direct solve, $\approx 3 \times 10^9$ FLOP per time step

34. In 3D the savings are even more significant.

35. For example, in the shuttle analysis: (20×10^6 grid points)

    (a) $\approx 10^{18}$ FLOP per time step for direct solve

    (b) $\approx 10^{10}$ FLOP per time step for the 3 block tridiagonals solves

The Delta Form

1. There are many ways can be devised to split the matrices and generate factored forms.
2. An especially useful form for ensuring a correct steady-state solution in a converged time-march: "delta form,"
3. Consider the unfactored form of the trapezoidal method

$$\left[I - \frac{1}{2}hA_x - \frac{1}{2}hA_y\right]U_{n+1} =$$

$$\left[I + \frac{1}{2}hA_x + \frac{1}{2}hA_y\right]U_n + h(\vec{bc}) + O(h^3) \quad (31)$$

4. From both sides subtract

$$\left[I - \frac{1}{2}hA_x - \frac{1}{2}hA_y\right]U_n$$

5. Define:$\Delta U_n = U_{n+1} - U_n$, and rewritting

$$\left[I - \frac{1}{2}hA_x - \frac{1}{2}hA_y\right]\Delta U_n = h\left[A_{x+y}U_n + (\vec{bc})\right] + O(h^3) \tag{32}$$

6. Note: the right side of this equation is the product of h and a term that is identical to the right side of our original ODE.

7. Thus, if Eq. 32 converges, it is guaranteed to converge to the correct steady-state solution of the ODE.

8. Factor Eq. 32 maintaining second-order accuracy

$$\left[I - \frac{1}{2}hA_x\right]\left[I - \frac{1}{2}hA_y\right]\Delta U_n = h\left[A_{x+y}U_n + (\vec{bc})\right] + O(h^3) \tag{33}$$

9. This is the delta form of a factored, second-order, 2-D equation.

10. In spite of the similarities in derivation between the non- "delta" and "delta" form, the convergence properties are vastly different.